

Künstl Intell (2011) 25: 73–74
DOI 10.1007/s13218-010-0074-7

DISSERTATIONEN UND HABILITATIONEN

Heuristic Evaluation Functions for General Game Playing

James E. Clune

Published online: 14 January 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract This dissertation explores the problem of constructing an effective general game-playing program, with an emphasis on techniques for automatically constructing effective heuristic evaluation functions from game descriptions. A technique based on abstract models of games is presented. The abstract model treats mobility, payoff and termination as the most salient elements of a game. Each of these aspects are quantified in terms of stable features. Evidence is presented that the technique produces heuristic evaluation functions that are both comprehensible and effective. Empirical work includes a series of general game-playing programs that placed first or second for the three consecutive years of the AAAI General Game-Playing Competition.

The full dissertation can be downloaded at <https://sites.google.com/site/jimcluneresearch/>.

1 Introduction

My work on general game playing techniques emphasizes heuristic evaluation functions. These are functions from game states to numbers used to assess the desirability of non-terminal states for particular players. In my opinion, automatic construction of effective heuristic evaluation functions from game descriptions is the central challenge of general game playing. However, it is a subject that is best dealt with in the context of a complete general game player employing some type of game-tree search. In recognition of this, search techniques are covered in this dissertation as well.

In my dissertation I describe several reasons why general game playing (GGP) is an important research problem.

Competitive Performance Metric In his doctoral dissertation, Barney Pell argues that a primary reason for computer game playing research has been the competitive performance metric for intelligence, namely the presumed link between winning games and intelligent behavior. Pell describes a problem with this presumption:

Unfortunately, the use of such a link has proved problematic: we have been able to produce strong programs for some games through specialized engineering methods, the extreme case being special-purpose hardware, and through analysis of the games by humans instead of by programs themselves. Consequently, it now appears that increased understanding and automation of intelligent processing is neither necessary nor sufficient for strong performance in game-playing [2].

Pell goes on to propose that the solution to this is to change the problem so that the programs are forced to do the game analysis because the programmer is not given the specific game to be played ahead of time. He calls this new paradigm *metagame*, which describes the basic ideas of what is now known as GGP.

Flexible Software Capabilities Machines in general and computers in particular are extremely brittle compared to the flexibility of behavior and abilities we see in the animal kingdom. Part of this brittleness is attributable to the extreme narrowness of domain that is associated with most software. The notion of domain breadth is difficult to characterize precisely. A single program can easily play top-quality chess, checkers, and Othello by simply combining the top

J.E. Clune (✉)
University of California, Los Angeles, USA
e-mail: jim.clune@gmail.com

programs for each game with a simple “switch” statement. However, this approach does not produce the ability to effectively play an infinite number of games that qualitatively differ from one another. GGP presents a challenge to exhibit flexibility in software that is more common to natural organisms than to computers.

Game-Oriented Programming Another way to view GGP is to think of the game descriptions as computer programs and the general game player as a compiler. When viewed through this lens, the present research program can be seen as an exploratory step in the development of a game-oriented programming paradigm. The paradigm enables a programmer to specify a task that can be formulated in terms of goals, actions, states, payoffs, and adversaries without explicitly supplying the compiler with any strategies or tactics for maximizing the payoff.

Generality as a Key Aspect of Intelligence A primary scientific problem of AI is to understand intelligence computationally. Our concern is with computational theories that circumscribe the design space of intelligent systems rather than psychological theories of human cognition. Nevertheless, because humans exhibit the most perspicuous example of intelligence, it is reasonable to strive for theories that capture features that have been identified as important to understanding human intelligence. Chief among these features is what psychologists call *g*, or the *general factor* of intelligence [1]. GGP research contributes to the scientific goals of AI by explicitly addressing generality as an important aspect of intelligence.

2 Overview of the Dissertation Project

The project associated with this dissertation is aimed at developing effective techniques for general game playing, emphasizing heuristic evaluation functions. My technique for heuristic evaluation functions involves the construction of an abstract model from a game description. The model is in terms of stable (incrementally varying) numeric features, and abstracts the game to core components, including payoff, mobility, and termination. The overall heuristic evaluation function seeks to pursue mobility and expected payoff,

weighing payoff more heavily as the game approaches termination.

An important aspect of the project is that these techniques are developed and evaluated in the context of a complete general game playing program. This project has co-evolved with the introduction of the AAAI General Game Playing Competition and the associated Stanford General Game Playing Project. To a first degree of approximation, my project can be understood as a series of entries in this competition. This does not mean that the value of my project is limited to the competition, but rather that my project represents one approach to the type of research that the competition is designed to promote. On a practical level, the Game Description Language (GDL) developed by the Stanford group has provided the model for which games are described in my own project, allowing me to focus more on game-playing techniques instead of on how to describe games.

The competition itself has created a small but international community of researchers as well as a venue for empirically evaluating various approaches. At the time of this writing, there have been three annual GGP competitions. My project has contributed an entrant in each competition and has consistently earned a spot in the championship round, placing first the first year and second the other two years. The level of game-play has increased significantly each year. At the time of this writing, my project is the only one to make it to the championship round more than once.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Jensen A (1999) The *g* factor: the science of mental ability. *Psychology* 10(23)
2. Pell B (1993) Strategy generation and evaluation for meta-game playing. PhD thesis, Trinity College, University of Cambridge

James E. Clune is a quantitative researcher/software developer at Jane Street Capital in New York City. He received his Ph.D. in computer science from UCLA in 2008 and B.S. in engineering from Harvey Mudd College in 1994. Prior to joining Jane Street Capital, he worked for Parasoft for ten years developing software tools.